

VZ200

YAHTZEE

This is a VZ200 version for the dice game Yahtzee, designed for an unlimited number of players.

Each player throws his or her dice up to three times each turn. After the first and second throws you can hold any dice you wish to keep, re-throwing the balance. After the third throw you must enter your score in the table provided.

Once a score has been recorded for a particular category, that category can't be used again. The game ends after 13 rounds.

Because of the limitations of the printer used to produce the listing it's wise to include the graphics [shift Js] in lines 2020 and 2050. The sections underlined should be inserted in inverse text.

The program occupies 5.9 Kbytes of memory.

Ian Thompson,
Collaroy Plateau, NSW

Main Variable Used

NS()	Player's name
S1\$()	Titles of category
S2\$()	Description of category
SC\$()	Update score for each category
DF()	Spots on dice
SC()	Score
ND()	Random number for dice
NP	Number of players
IP	User update of scoresheet
TURN	Turn number

Main Routines

0- 46	Title graphics
100	Initialises screen background
	Clears memory for variables
130	Input players' names
140	Initialises variables
160-170	Input players names
165	Limits player's name to 11 characters
180-190	Set number of player turns per number of players
210	Random number generator for dice throw
290-320	Print score table
330-420	User update of score
470-570	Updates total score
600-650	Displays final score and placings
1000-1120	Data statements for score table
1130-1160	Data statements for spots on dice
2020-2050	Displays dice
9000-20040	User update of score subroutine
21000-22140	Instructions

```

0 *****
1 '* VZ-200 Y A H T Z E E *
2 '* IAN THOMPSON - COLLAROY *
3 *****
4 CLS:SOUND 25,6:COLOR,0
5 FOR X=1 TO 32:POKE 28671+X,204:POKE 29151+
X,195
6 POKE 28672,174:POKE 28703,173:POKE 29152,1
71:POKE 29183,167
7 NEXT X
8 FOR N=28704 TO 29120 STEP 32
9 POKE N,202
10 NEXT N
11 FOR O=28735 TO 29151 STEP 32
12 POKE O,197
13 NEXT O
22 PRINT@106," YAHTZEE "
24 A$=" IAN A.THOMPSON "
26 B$="COLLARROY PLATEAU"
28 FOR N=1 TO LEN(A$)
30 PRINT@231,RIGHT$(A$,N);
32 PRINT@263,RIGHT$(B$,N);
34 NEXT
35 FOR I=1 TO 500:NEXT I
36 PRINT@454,"INSTRUCTIONS (Y/N)?"
38 IF INKEY$<>" " THEN 38
40 A$=INKEY$
42 IF A$="N" THEN SOUND30,1:GOTO100
44 IF A$<>"Y" THEN 40
45 SOUND30,1
46 GOSUB 21000:'INSTRUCTIONS
100 POKE 30744,0:COLOR 5,0:CLEAR 1000
120 R=RND(0)
130 CLS:PRINT@128,"NO. OF PLAYERS";:INPUT NP
135 SOUND 31,1
140 DIM SC$(13,NP),S1$(13),S2$(13),N$(NP),DF
(6,6),SC(NP),YF(NP)
145 GOSUB 1000
150 FOR I=1 TO NP
160 CLS:PRINT@128,"PLAYER #";I;:INPUT "S NAM
E";N$(I)
162 SOUND 31,1
165 IF LEN(N$(I))>11 THEN SOUND 20,1;10,1:GO
TO 160
170 NEXT
180 FOR TURN = 1 TO 13
190 FOR PL=1 TO NP
210 FOR R=1 TO 5:ND(R)=RND(6):NEXT
220 GOSUB 2000
230 GOSUB 3000
240 PRINT@416,"REMEMBER THESE,THEN"
250 PRINT"HIT ANY KEY TO CONTINUE":A$=INKEY$
260 A$=INKEY$:IF A$="" THEN 260
270 FOR I=1 TO 6:N(I)=0:NEXT
280 FOR I=1 TO 5:N(ND(I))=N(ND(I))+1:NEXT
290 CLS:PRINT"CHOOSE A CATEGORY,";N$(PL)
300 FOR I=1 TO 13:PRINTUSING"##" ";I;
310 PRINTS1$(I);S2$(I);SC$(I,PL)
320 NEXT
330 PRINT:INPUT"WHICH [1-13]";:IF
340 IF IP<1 OR IP>13 THEN 290
345 IF IP=12 THEN 12000
350 IF SC$(IP,PL)<>" " THEN 15000
360 IF IP<7 THEN SC$(IP,PL)=STR$(IP*N(IP))
370 IF IP=7 OR IP=8 THEN 7000
380 IF IP=9 THEN 9000
390 IF IP=10 THEN 10000
400 IF IP=11 THEN 11000
420 IF IP=13 THEN 13000

```


POCKET PROGRAMS

VZ200

```

430 NEXT PL:NEXT TURN
440 FOR PL=1 TO NP:FOR I=1 TO 13
460 NEXT
470 FOR I=1 TO 6
480 SC(PL)=SC(PL)+VAL(SC$(I,PL)):NEXT
490 IF SC(PL)>62 THEN SC(PL)=SC(PL)+35
500 FOR I=7 TO 13
510 SC(PL)=SC(PL)+VAL(SC$(I,PL))
520 NEXT:NEXT
530 FOR I=1 TO NP-1
540 HI=0:FOR J=1 TO NP
550 IF SC(J)>HI THEN HI=SC(J):P=J
560 NEXT
570 D=SC(I):SC(I)=SC(P):SC(P)=D
580 D$=N$(I):N$(I)=N$(P):N$(P)=D$
585 NEXT
590 SOUND 20,1:SOUND 10,1:SOUND 20,1
600 CLS:PRINT"AND THE PLACINGS ARE";
620 PRINT:PRINT:FOR I=1 TO NP
630 PRINTUSING"###";I;
640 PRINT" ";N$(I):TAB(25);SC(I)
650 NEXT
660 PRINT@480,"ANOTHER GAME (Y/N)?";
670 GOSUB 20000
680 IF YN$="Y" THEN RUN
690 CLS:PRINT@162,"THANKS FOR THE GAME - BYE
":END
1000 DATA STATEMENTS
1040 FOR I=1 TO 13
1050 READ S1$(I),S2$(I):NEXT
1060 DATA"ACES",".", [SUM OF 1'S] "-", "TWO'S",
", [SUM OF 2'S] -"
1070 DATA"THREES",".", [SUM OF 3'S] "-", "FOURS",
", [SUM OF 4'S] -"
1080 DATA"FIVES",".", [SUM OF 5'S] "-", "SIXES",
", [SUM OF 6'S] -"
1090 DATA"3 OF A KIND",".", [SUM] "-", "4 OF A
KIND",".", [SUM] -"
1100 DATA"FULL HOUSE",".", [25] "-", "SM. STR
AIGHT",".", [30] -"
1110 DATA"LG. STRAIGHT",".", [40] "-", "YAHTZEE
",".", [50] -"
1120 DATA"CHANCE",".", [SUM] -"
1130 FOR I=1 TO 6:FOR J=1 TO I
1140 READ DF(I,J):NEXT:NEXT
1150 DATA 66,33,99,1,66,131,33,35,97,99
1160 DATA 1,3,66,129,131,1,3,65,67,129,131
1190 RETURN
2000 CLS:PRINT N$(PL):"S ROLL"
2010 FOR R=96 TO 224 STEP 32:FOR S=2 TO 26 S
TEP 6
2015 COLOR 5
2020 PRINT@R+S," ";THREE SHIFT J'S
2030 NEXT:NEXT
2040 FOR D=1 TO 5:FOR N=1 TO ND(D)
2045 COLOR 3
2050 PRINT@91+D*6+DF(ND(D),N)," ";ONE SHIFT
J
2060 NEXT:NEXT
2070 RETURN
3000 FOR K=1 TO 2:F=1
3010 FOR J=1 TO 5
3020 P=252+J*6:RR(J)=0
3030 PRINT@P,"^^^^"
3035 PRINT:PRINT
3040 PRINT"REROLL THIS ONE [Y/N]?"
3050 PRINT"[S FOR SCOREBOARD]":PRINT"[M FOR
MISTAKE]":YN$=INKEY$
3060 YN$=INKEY$
3070 IF YN$<>"Y" AND YN$<>"N" AND YN$<>"S" A

```

```

ND YN$<>"M" THEN3060
3072 IF YN$="Y" THEN SOUND 20,1
3074 IF YN$="N" THEN SOUND 10,1
3076 IF YN$="S" THEN SOUND 15,1
3078 IF YN$="M" THEN SOUND 20,2;10,1
3080 IF YN$<>"S" THEN 3130
3090 CLS:PRINT TAB(5);N$(PL):"S SCORES"
3100 FORI=1TO13:PRINTUSING"###";I;
3105 PRINT S1$(I);S2$(I);SC$(I,PL)
3110 NEXT:PRINT"HIT ANY KEY TO RETURN:":A$=I
NKEY$
3120 A$=INKEY$:IF A$="" THEN 3120 ELSE GOSUB
2000:GOTO 3020
3130 PRINT@P," "
3140 IF YN$="M" THEN 3010
3150 IF YN$="Y" THEN RR(J)=1
3160 NEXT
3170 FOR I=1 TO 5:IF RR(I)=1 THEN ND(I)=RND(
6):F=0
3180 NEXT:IF F THEN K=2
3190 GOSUB 2000
3200 NEXT
3210 RETURN
7000 FOR I=1 TO 6:IFN(I)>IP-5 THEN 7030
7010 NEXT
7020 GOTO 16000
7030 SC=0
7040 FOR I=1 TO 5:SC=SC+ND(I):NEXT
7050 SC$(IP,PL)=STR$(SC)
7060 GOTO 430
9000 FOR I=1 TO 6
9010 IF N(I)>2 THEN N(I)=N(I)-3:GOTO 9040
9020 NEXT
9030 GOTO 16000
9040 FOR I=1 TO 6
9050 IF N(I)>1 THEN 9080
9060 NEXT
9070 GOTO 16000
9080 SC$(9,PL)=" 25"
9090 GOTO 430
9090 GOTO 430
10000 FOR I=1 TO 3:F=1:FOR J=1 TO I+3
10010 IF N(J)=0 THEN F=0
10020 NEXT
10030 IF F THEN 10060
10040 NEXT
10050 GOTO 16000
10060 SC$(10,PL)=" 30"
10070 GOTO 430
11000 FOR I=1 TO 2:F=1:FOR J=1 TO I+4
11010 IF N(J)=0 THEN F=0
11020 NEXT
11030 IF F THEN 11060
11040 NEXT
11050 GOTO 16000
11060 SC$(11,PL)=" 40"
11070 GOTO 430
12000 FOR I=1 TO 6
12010 IF N(I)=5 THEN 12040
12020 NEXT
12030 GOTO 16000
12040 SC$(12,PL)=" 50"
12050 IF YF(PL) THEN SC$(12,PL)=STR$(VAL(SC$
(12,PL))+100)
12060 YF(PL)=1
12070 GOTO 430
13000 SC=0:FOR I=1 TO 5
13010 SC=SC+ND(I):NEXT
13020 SC$(13,PL)=STR$(SC)
13030 GOTO 430

```


VZ200

```

15000 SOUND 15,1:CLS
15010 PRINT@128,"YOU'VE ALREADY DONE"
15020 PRINT"THE ";S1$(IP); " ";N$(PL)
15030 FOR I=1 TO 2000:NEXT
15040 GOTO 290
16000 SOUND 15,1:CLS
16010 PRINT@128,"YOU'RE NOT ELIGIBLE FOR"
16020 PRINT"A ";S1$(IP); " ";N$(PL)
16025 IF IP=12 AND YF(PL) THEN SOUND 0,8:GOT
0 290
16030 PRINT:PRINT:PRINT"DO YOU WANT IT ANYWA
Y [Y/N]?";
16040 GOSUB 20000
16050 IF YN$="N" THEN 290
16060 SC$(IP,PL)=" 0"
16070 IF IP=12 THEN YF(PL)=1
16080 GOTO 430
20000 YN$=INKEY$
20010 YN$=INKEY$:IF YN$="" THEN 20010
20020 IF YN$<>"Y" AND YN$<>"N" THEN 20000
20030 IF YN$="Y" THEN SOUND 20,1 ELSE SOUND
10,1
20040 RETURN
21000 CLS:PRINT"INSTRUCTIONS"
21010 PRINT:PRINT"IN THIS DICE GAME EACH PLA
YER"
21020 PRINT"CAN THROW UP TO THREE TIMES EACH
";
21030 PRINT"TURN. AFTER THE FIRST THROW, HE"
21040 PRINT"CAN SET ASIDE ANY DICE HE WISHES
";
21050 PRINT"TO KEEP,AND RETHROW THE BALANCE.
";
21060 PRINT"HE CAN DO THE SAME AFTER THE"
21070 PRINT"SECOND AND THIRD THROWS. HE CAN,
";
21080 PRINT"OF COURSE, STOP BEFORE THE THIRD
";
21090 PRINT"THROW IF HE WISHES."
21100 PRINT"ONCE THE PLAYER HAS DECIDED TO"
21110 PRINT"STOP, HE MUST DECIDE INTO WHICH"
21120 PRINT"CATEGORY TO ENTER HIS SCORE."
21130 GOSUB 22100
21140 DIM S1$(13),S2$(13)
21150 FOR I=1 TO 13
21152 READ S1$(I),S2$(I):NEXT
21154 CLS
21156 PRINT
21160 FOR I=1 TO 13
21170 PRINT S1$(I):S2$(I):NEXT
21172 PRINT@300,"[SUM OF HOUSE] -"
21174 PRINT@334,"[1,2,3,4,5,] -"
21176 PRINT@366,"[2,3,4,5,6,] -"
21178 PRINT@394,"[FIVE OF A KIND] -"
21180 PRINT@427,"[ANY FIVE DICE] -"
21200 GOSUB 22100
21210 CLS:PRINT"THE GAME ENDS AFTER 12 ROUND
S."
21220 PRINT"ONCE A SCORE HAS BEEN RECORDED"
21230 PRINT"FOR A PARTICULAR CATEGORY, THAT"
21240 PRINT"CATEGORY CAN'T BE USED AGAIN."
21250 GOSUB 22100
21260 RETURN
22100 PRINT@485,"PRESS <C> TO CONTINUE";
22110 IF INKEY$<>"C" THEN 22100
22120 IF INKEY$="" THEN 22100
22130 IF INKEY$="C" THEN 22130
22140 SOUND 30,1:RETURN

```

Most BASICs

SORTING OUT
THE SORTS

The program listed with this article was developed to test the speed and efficiency of four sorting algorithms: Insertion, Shell, Quick and Selection. The program will probably run as is in most BASICs.

What's usually required of a sort is to put a list of names into alphabetical order, but the average textbook seems to present sorts for numbers with no indication of the best choice for a particular task.

The choice of sorting algorithms is broad: more than three dozen are known, spread across some hundred texts. The most popularly presented, and the slowest if list size is more than 11, is the bubble sort.

Fortunately, the choice can be narrowed down to short algorithms which work in RAM only and are not bubbly sorts in disguise!

The program

The first program line, line 30, asks for the size of the array to be generated and sorted. Start with a choice of 10 to check the sorts are working as expected.

Lines 70-130 generate the required number of capital letters and place them in the array. The variable CD is set at 64, one less than the ASCII code for 'A'. A number is generated in line 90 and added to CD. If this number is acceptable, the character it represents is placed in array CH\$. If not, C is decremented by one and the process is repeated. The loop runs until BN (big number!) letters are placed in the array. Line 130 prints out the unsorted list and could be omitted from the program.

The straight Insertion sort and the Shell (insertion algorithm) sort are particularly useful.

The Selection sort is always the slowest, since the same number of compares and swaps is made if the list is random, or if only one element is out of order. The Quick sort isn't much better if only a few